As pointed out during the recent interview (for which the undersigned thanks the Examiner), **Gunyakti et al.** does not teach generating a PKI certificate for each software component, nor does Gunyakti use generated PKI certificates to code sign each of the different executable software components within each gaming machine.

Instead, Gunyakti et al. generates a volume license for a number of products and teaches that this volume license may then be signed with a private key to generate the license file 224 – see paragraph [0027]:

> VLK **222** may be embedded anywhere in the data, which in one embodiment is then signed with a private key to generate the license file **224**. In one embodiment of the

Therefore, in Gunyakti, it is the license to use the software that is signed, and not the software components themselves, as in the claimed embodiments.

This allows Gunyakti to verify the integrity of the license file 224:

> **[0029]** The integrity of license file **224** can be verified by checking the signature of the file. If the license file **224** has not been tampered with, i.e., if the integrity is verified, the software is allowed to run. If the license file **224** has been tampered with, the software will run in a reduced functionality mode.

However, in Gunyakti, it is the license file 224 that is signed, and not the software components to which the license refers. As pointed out during the interview, this distinction should not be overlooked. The Office presents factually incorrect grounds for its §103(a) rejection

on page 3 of the Final Office Action and again on pages 8-9 in the "**Response to Argument**" section when it states that Gunyakti's paragraphs 0026-0028 teach to code sign executable software components. Indeed, Gunyakti's license file 224 is code signed, and not the underlying software components.

Next, the Office relies upon **Yip** for the same teaching of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, and points to Figs. 2 and 3 and paragraphs 0048 and 0046.

In Yip, a conventional Certificate Authority (CA) issues a certificate 106 and an application-specific CA issues a corresponding application-specific certificate 206. See paragraph 0042. The certificate 106 and application certificate 206 are linked, such that when the certificate 106 is revoked, the application-specific certificates are also preferably revoked. See paragraph 0044.

Thus, the application-specific certificate 206 is a "companion" to the certificate 106 (note error wherein second instance of "106" in the passage below should be "206"):

> **[0046]** Thus, for every certificate **106** issued by the CA **104**, a "companion," application-specific certificate **106** is issued by the application-specific CA **204** for use with the particular application **201**. Advantageously, the format of the

Note, however, that claim 17 recites:

> **code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified at least**

> **by a unique identifier that is uniquely associated with the executable software component** ***such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate,*** **... (italics for emphasis)**

As the application-specific certificate 204 is "for use with the **particular application 201**", it follows that identical executable software components in different ones of the plurality of gaming machines, in Yip, would be associated with different PKI certificates, as each application (each "**particular application 201**", in Yip's language) would receive a different certificate 106 and corresponding different application-specific certificates 206. There is no teaching or suggestion in Yip otherwise.

Indeed, Yip teaches away from the claimed embodiments in which **identical application-specific certificates** are provided for identical executable software components in different machines. In other words, the CA in Yip would not issue identical certificates 106 to more than one user/user nor would the CA issue identical companion application-specific certificates 206 to more than on user/machine, as each certificate 106 is different and as the application-specific certificates 206 are companions to such different certificates 106.

Therefore, since each "particular" application 201 receives a different certificate in Yip, there are believed to be no grounds for holding that Yip teaches or suggests (either alone or in combination with any or all of the other three applied references), the claimed limitation:

identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates

The applied reference to **Fieres** teaches the issuance of application certifications to insure that applications operate at the proper cryptographic level granted for that application by an application domain authority 22. However, there is no teaching or suggestion in Fieres that:

"identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates"

Nor is there any teaching or suggestion in Fieres that

"non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates"

... as also claimed. Fieres does not teach or suggest that "no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate", as claimed herein – nor has the Office identified where such teachings or suggestions may be found. In fact, it is highly unlikely that, in the context of the distribution of cryptographic capabilities, that Fieres would allow identical executable components in different machines to have identical certificates, as required herein. Such would surely defeat the security measures. A general allegation that Fieres teaches application certificates with application IDs does not, without more, rise to the level of teaching the aforementioned claim limitations, whether considered singly or in combination with the other applied references.

Lastly, **Lambert** was relied on for an alleged teaching of a software restriction policy certificate rule for each of the plurality of executable software components. However, Lambert does not teach a software restriction policy certificate rule for each executable software component. Quite to the contrary, Lambert teaches **one rule for an entire security level** for executing executable software (see Abstract, lines 3-4). This means that executable software, in Lambert et al. are associated with different security levels, which may allow or disallow execution thereof. Lambert also teaches a hierarchy of rules, to help distinguish which rule to use should a piece of software having multiple classifications (see Abstract, last sentence). In Column 15, Lambert teaches how rules are selected...

> the enforcement mechanism **518** can locate a rule from the signature, path information, or zone information associated with the file **510**. Note that while FIG. **5**A essentially represents accessing the policy to get the rule or rules via arrows labeled seven (7) and eight (8), the policy may be consulted more than once, e.g., to look first for a rule for the hash value, and if not found, for a rule for a signature (if any), and so on. Note that as described below with respect

...and how rules determine the execution of the file.

> nism (circled numeral two (2) in FIG. **5**A). As described below, based on this information the enforcement mechanism consults the effective policy **502** to determine which rule applies for the file **510**, and from the rule determines whether to open/execute the file, and if so, the extent of any restricted execution context for the file **510**.

In Lambert, therefore, there is no on-to-one relationship (a SRP for each executable software components) with executable software components and rules, as required by claim 17:

> **configuring a software restriction policy certificate rule for each of the plurality of executable software components and enforcing each of the**

software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized.

To the contrary, Lambert et al. teach a one-to-many relationship between the security rules and the executable software components, which is antithetical to the claimed embodiments, which require a software restriction policy for EACH of the plurality of executable software components.

Also, Lambert et al. do not provide any further teaching or suggestion, whether considered alone or in combination with Gunyakti, Yip or Fieres, of the limitation:

code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified at least by a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate, and

Considering now the applied references in combination, the undersigned notes that application-specific certificates exist, as taught by Fieres and Yip. However, the applied combination still fails to teach or to suggest:

producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, each software component subject to receiving certification including a unique identifier;

or

> **code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified at least by a unique identifier that is uniquely associated with the executable software component *such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate*, ... (italics for emphasis)**

or

> **configuring a software restriction policy certificate rule for each of the plurality of executable software components and enforcing each of the software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized.**

... as claimed in independent claim 17. In fact, the Office's primary reference to Gunyakti teaches volume licenses (the antithesis of the claimed embodiment) and Yip teaches "companion" certificates tied to (and revoked along with) a conventional certificate. Moreover, Lambert teaches a one-to-many relationship between the rules and the applications, such that the correct rule must be determined before execution of the application is allowed. Not only are the claimed elements absent from the applied combination or any of the individual references, but the combination as a whole appears to teach away from an embodiment that include separate and unique PKI certificates, the code signing step or configuring SRPs for each executable software component, as Gunyakti teaches the exact opposite, as Yip teaches that the conventional and application-specific

certificates are linked together (one is the "companion" to the other) and as Lambert teaches a one-to-many relationship between the security rules and the applications.

Claim 20 recites:

> **code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;**
>
> **configuring a separate software restriction policy for each authorized software component in each of the constituent computers of the gaming system, and associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed;**
>
> **enforcing the associated software restriction policy for each code signed authorized software component such that each code signed authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy.**

The arguments presented above relative to claim 17 are equally applicable to claim 20. Rather than repeat these here, reference is made to the arguments above, which are incorporated herein in their entirety, as if repeated here in full.

Claim 22 was rejected as being unpatentable over Lambert-Gunyakti-Yip.

Claim 22 recites, similarly to claims 17 and 20:

> **configuring a separate and unique certificate software restriction policy for each authorized executable software component of each of the constituent computers of the gaming system such that the each authorized executable software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy;**

> **code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;**

Although Lambert does teach rules based upon a path in Column 13, the applied combination of Lambert-Gunyakti-Yip does not teach or suggest the configuring and code signing steps recited above, nor, by extension, the claimed step of:

> **enforcing the certificate software restriction policy configured for each of the code signed authorized executable software components of each of the constituent computers of the gaming system, and**

for the same arguments as were presented above relative to claim 17. These same arguments are incorporated herein in their entirety, as if repeated here in full.

Claim 24 recites:

> **producing a separate and unique PKI certificate for each of the plurality of executable software components within the gaming system subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate;**

... for which the arguments above are applicable.

Moreover, claim 24 also recites:

code signing each software component subject to receive certification with its respective separate and unique PKI certificate;

configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates, and

enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates.

In contradistinction, the primary reference to Gunyakti advocates volume licenses, Yip advocates companion application-specific certificates and Lambert calls for a hierarchy of rules to enable the application of a specific rule to a specific application. The applied combination does not teach code signing each software component subject to receive certification with its respective separate and unique PKI certificate (compare to Gunyakti's volume licenses), configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates (compare with the one-to-many relationship of Lambert's rules to the applications) or enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates, as claimed herein.

Independent claim 25, similarly to the claims above, recites:

for each of the plurality of gaming machines of the network connected gaming system:

code signing each authorized executable software component with a separate PKI certificate that is unique to the authorized software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different gaming machines are code signed with a same PKI certificate;

... and the arguments advanced hereinabove are incorporated by reference.

However, claim 25 also recites:

> **packaging the code signed authorized software components into an installation package;**
>
> **configuring install policies to install each code signed authorized executable software component contained in the installation package;**
>
> **configuring certificate rule policies to allow execution of the installed code signed authorized executable software component;**
>
> **configuring enforcement of the policies.**

➔ **Although claim 25 includes different limitations than claim 17, the Office once again in its Final Rejection of November 24, 2009 limited its examination of this claim to a statement that it "encompasses limitations that are similar to claim 17" and rejected the claim on the same rationale. This is second time that the Office has failed to substantively examine claim 25.**

This alone, it is respectfully submitted warrants withdrawal of the finality of the outstanding Office Action and the issuance of either a new non-final Office Action, or a Notice of Allowance, as appropriate.

However, as the Office will note, claim 25 also includes a recitation of "packaging the code signed authorized software into an installation package", which finds no counterpart in any of the previous independent claims and no counterpart in any of the applied references, whether considered singly or in combination. Similarly, the claim also calls for configuring install policies and for configuring enforcement of the policies. The applied combination does not teach or suggest any such embodiment, nor has the office pointed to any such teachings in the applied combination.

In the "**Response to Arguments**" section, the Office cited KSR and stated "[t]he combination of familiar elements according to known methods is likely to be obvious when it does not more than yield predictable results". However, this is not a case of familiar elements being combined according to known methods. Indeed, the Office cited four different references, and not one of these references teaches or suggests, for example,

> **"... identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate"**

Indeed, it is believed to be counter-intuitive for identical executable software components in different machines be code signed with identical PKI certificates. However, it does makes sense in the context of Casino gaming, in which estates of hundreds or thousands of identical or near identical gaming machines run the same regulated and gaming-jurisdiction certified software.

None of the applied references, alone or in combination, teach or suggest the claimed embodiments. The prior art (Yip) teaches that each "particular" software is signed with a different certificate. The prior art (Gunyakti) also teaches code signing volume licenses (not executable software components). The prior art also teaches security levels (Lambert) or cryptographic levels (Fieres) that may or may not allow execution of software components. It is respectfully submitted that the claimed elements are most assuredly not combined "according to known methods", as the Office asserts in its attempt to apply KSR to the pending claims. There is not believed to be any